What is claimed is:

sub A1

~~1. A processor for reading instructions from a memory according to a program counter, the memory storing instructions in one-byte units, and for executing the read instructions,~~

~~the program counter including a first program counter and a second program counter,~~

~~the first program counter indicating a storage position of a processing packet in the memory, the processing packet being composed of an integer number of the one-byte units,~~

~~the second program counter indicating a position of processing target instruction in the processing packet, the processing target instruction being an operation to be executed by the processor.~~

Sub C7

~~2. The processor of Claim 1, including a first program counter updating means and a second program counter updating means,~~

~~the second program counter updating means incrementing a value of the second program counter in accordance with an amount of instructions that were executed in a preceding cycle and sending any carry generated in an incrementing to the first program counter updating means, and~~

~~the first program counter updating means adding the carry received from the second program counter updating~~

Sub C7

means to the value of the first program counter.

3. The processor of Claim 2, further including:

program counter relative value extracting means for extracting, when an instruction being executed includes a program counter relative value that is based on an address of a first instruction executed in a present cycle, the program counter relative value; and

calculating means for adding the program counter relative value to the value of the first program counter and the value of the second program counter, and setting an addition result as the value of the first program counter and the value of the second program counter.

4. The processor of Claim 3,

wherein the calculating means includes a first calculating unit and a second calculating unit,

the second calculating unit adding the value of the second program counter and lower bits of the program counter relative value, setting a result of an addition as the value of the second program counter, and sending any carry generated in the addition to the first calculating unit,

the first calculating unit adding the value of the first program counter, upper bits of the program counter relative value, and any carry received from the second calculating unit, and setting a result of an addition as

Sub C'

the value of the first program counter.

5. The processor of Claim 3, wherein the calculating means includes a first calculating unit and a second calculating unit,

the second calculating unit adding the value of the second program counter and lower bits of the program counter relative value without generating a carry, and setting a result of an addition as the value of the second program counter,

the first calculating unit adding the value of the first program counter and upper bits of the program counter relative value, and setting a result of an addition as the value of the first program counter.

6. The program counter of Claim 3, wherein the calculating means adds the value of the first program counter and upper bits of the program counter relative value, sets a result of an addition as the value of the first program counter, and sets lower bits of the program counter relative value as the value of the second program counter.

7. The processor of Claim 3, wherein the calculating means adds the program counter relative value and a value whose upper bits are the value of the first program counter and lower bits are the

Sub C'7

value of the second program counter, and sets upper bits of a result of an addition as the value of the first program counter and lower bits of the result as the second program counter.

8. The processor of Claim 2, further including:

program counter relative value extracting means for extracting, when an executed instruction includes a program counter relative value that is based on an address of the executed instruction, the program counter relative value;

program counter amending means for amending the value of the first program counter and the value of the second program counter to indicate an address of the executed instruction; and

calculating means for adding the program counter relative value, the value of the first program counter, and the value of the second program counter, and setting a result of an addition as the value of the first program counter and the value of the second program counter.

9. The processor of Claim 2, further including:

program counter relative value calculating instruction decoding means for decoding a program counter relative value calculating instruction that performs an addition using a program counter relative value and one of

(a) a value of the program counter stored in a register, and

Sub C7

~~(b) the value of the first program counter and the value of the second program counter;~~

~~calculating means for performing the addition indicated by the program counter relative value calculating instruction to generate an addition result; and~~

~~program counter value updating means for storing the addition result in one of~~

~~(a) the register, and~~

~~(b) the first program counter and the second program counter.~~

Sub B2

~~10. The processor of Claim 1,~~

~~wherein the first program counter indicates a memory address, the memory address being a storage position in the memory of a processing packet that is given by bit shifting the value in the first program counter by log_2n bits in a leftward direction, *n* being a length of a processing packet in bytes.~~

Sub A2

~~11. The processor of Claim 10, further including~~

~~an instruction buffer for temporarily storing instructions; and~~

~~instruction reading means for transferring instructions with a minimum transfer size of one one-byte unit from the memory to the instruction buffer, in accordance with available space in the instruction buffer but regardless of a size of a processing packet.~~

12. An instruction sequence optimizing apparatus, for generating optimized code from an instruction sequence, comprising:

address assigning means for estimating a size of each instruction in the instruction sequence and assigning an address to each instruction, upper bits of each address indicating a memory address at which a processing packet is stored and lower bits of each address indicating a processing target instruction in the processing packet;

label detecting means

(1) for detecting a label, which should be resolved by an address of a specified instruction, from the instruction sequence, ~~and~~ obtaining the address of the specified instruction, and

(2) for detecting a label, which should be resolved by a difference in addresses of two specified instructions, from the instruction sequence, and obtaining the addresses of the two specified instructions;

program counter relative value calculating means for calculating, when a label which should be resolved by a difference in addresses of two specified instructions has been detected, a program counter relative value by subtracting an address of one of the two specified instructions from an address of another of the two specified instructions;

converting means

(1) for converting an instruction that has a label that should be resolved by an address of a specified instruction into an instruction with a size that is based on a size of the address of the specified instruction,

(2) for converting an instruction that has a label that should be resolved by a difference in addresses of two specified instructions into an instruction with a size that is based on a size of the program counter relative value calculated from the addresses of the two specified instructions; and

optimized code generating means for generating optimized code by converting addresses of instructions in accordance with the sizes of instructions after conversion by the converting means.

13. The instruction sequence optimizing apparatus of Claim 12,

wherein the program counter relative value calculating means includes a lower bit subtracting unit and an upper bit subtracting unit,

the lower bit subtracting unit subtracting lower bits of the address of the one of the two specified instructions from lower bits of the address of the other of the two specified instructions, for setting a result of a subtraction as lower bits of the program counter relative value, and sending any carry generated in the subtraction to the upper bit subtracting unit, and

the upper bit subtracting unit subtracting upper bits of the address of one of the two specified instructions and any carry received from the lower bit subtracting unit from upper bits of the address of the other of the two specified instructions, and for setting a result of a subtraction as upper bits of the program counter relative value.

14. The instruction sequence optimizing apparatus of Claim 12,

wherein the program counter relative value calculating means includes a lower bit subtracting unit and an upper bit subtracting unit,

the lower bit subtracting unit subtracting lower bits of the address of one of the two specified instructions from lower bits of the address of the other of the two specified instructions without generating a carry and setting a result of a subtraction as lower bits of the program counter relative value, and

the upper bit subtracting unit subtracting upper bits of the address of one of the two specified instructions from upper bits of the address of the other of the two specified instructions, and for setting a result of a subtraction as upper bits of the program counter relative value.

15. The instruction sequence optimizing apparatus of Claim 12,

wherein the program counter relative value calculating means subtracts upper bits of an address of one of the two specified instructions from upper bits of an address of the other of the two specified instructions, sets a result of a subtraction as upper bits of the program counter relative value, and sets lower bits of the other of the two specified instructions as lower bits of the program counter relative value.

16. An assembler that generates relocatable code from an instruction sequence, each address of an instruction in the instruction sequence having upper bits that indicate a memory address at which a processing packet is stored and lower bits that indicate a position of processing target instruction that is included in the processing packet,

the assembler comprising:

label detecting means for detecting a label in the instruction sequence that should be resolved by a difference in addresses between two specified instructions, and obtaining the addresses of the two specified instructions;

program counter relative value calculating means for calculating a program counter relative value by subtracting an address of one of the two specified instructions from an address of another of the two specified instructions; and

replacing means for replacing the label with the program counter relative value calculated by the program counter relative value calculating means.

17. The assembler of Claim 16,

wherein the program counter relative value calculating means includes a lower bit subtracting unit and an upper bit subtracting unit,

the lower bit subtracting unit subtracting lower bits of the address of the one of the two specified instructions from lower bits of the address of the other of the two specified instructions, for setting a result of a subtraction as lower bits of the program counter relative value, and sending any carry generated in the subtraction to the upper bit subtracting unit, and

the upper bit subtracting unit subtracting upper bits of the address of one of the two specified instructions and any carry received from the lower bit subtracting unit from upper bits of the address of the other of the two specified instructions, and for setting a result of a subtraction as upper bits of the program counter relative value.

18. The assembler of Claim 16,

wherein the program counter relative value calculating means includes a lower bit subtracting unit and an upper bit subtracting unit, the lower bit subtracting unit subtracting lower bits of the address of one of the two specified instructions from lower bits of the address of the other of the two specified instructions without generating a carry and setting a result of a subtraction as lower bits of the program counter relative value, and

the upper bit subtracting unit subtracting upper bits of the address of one of the two specified instructions from upper bits of the address of the other of the two specified instructions, and for setting a result of a subtraction as upper bits of the program counter relative value.

19. The assembler of Claim 16,

wherein the program counter relative value calculating means subtracts upper bits of an address of one of the two specified instructions from upper bits of an address of the other of the two specified instructions, sets a result of a subtraction as upper bits of the program counter relative value, and sets lower bits of the other of the two specified instructions as lower bits of the program counter relative value.

20. A linker that generates object code by combining relocatable code, each address of an instruction in the relocatable code having upper bits that indicate a memory address at which a processing packet is stored and lower

bits that indicate a position of processing target instruction that is included in the processing packet,

the linker comprising:

relocation information detecting means for detecting a label in the relocatable code that should be resolved by a difference in addresses between two specified instructions, and obtaining the addresses of the two specified instructions;

program counter relative value calculating means for calculating a program counter relative value by subtracting an address of one of the two specified instructions from an address of another of the two specified instructions; and

replacing means for replacing the label with the program counter relative value calculated by the program counter relative value calculating means.

21. The linker of Claim 20,

wherein the program counter relative value calculating means includes a lower bit subtracting unit and an upper bit subtracting unit,

the lower bit subtracting unit subtracting lower bits of the address of the one of the two specified instructions from lower bits of the address of the other of the two specified instructions, for setting a result of a subtraction as lower bits of the program counter relative value, and sending any carry generated in the subtraction to the upper bit subtracting unit, and the upper bit subtracting unit subtracting upper bits of the address of one of the two specified instructions and any carry received from the lower bit subtracting unit from upper bits of the address of the other of the two specified instructions, and for setting a result of a subtraction as upper bits of the program counter relative value.

22. The linker of Claim 20,

wherein the program counter relative value calculating means includes a lower bit subtracting unit and an upper bit subtracting unit,

the lower bit subtracting unit subtracting lower bits of the address of one of the two specified instructions from lower bits of the address of the other of the two specified instructions without generating a carry and setting a result of a subtraction as lower bits of the program counter relative value, and

the upper bit subtracting unit subtracting upper bits of the address of one of the two specified instructions from upper bits of the address of the other of the two specified instructions, and for setting a result of a subtraction as upper bits of the program counter relative value.

23. The linker of Claim 20,

wherein the program counter relative value calculating means subtracts upper bits of an address of one of the two specified instructions from upper bits of an address of the other of the two specified instructions, sets a result of a subtraction as upper bits of the program counter relative value, and sets lower bits of the other of the two specified instructions as lower bits of the program counter relative value.

24. A disassembler that receives an indication of an address of an instruction in object code and outputs an assembler name of the instruction at the indicated address, each address of an instruction in the object code having upper bits that indicate a memory address at which a processing packet is stored and lower bits that indicate a position of processing target instruction that is included in the processing packet,

the disassembler comprising:

program counter relative value extracting means for extracting, when the indicated instruction includes a program counter relative value, the program counter relative value from the indicated instruction;

label addressing calculating means for adding an address of the indicated instruction to the extracted program counter relative value and setting an addition result as a label address;

storing means for storing a label name corresponding to each label address; and

searching means for searching the storing means for a label name that corresponds to the calculated label address and outputting the corresponding label name.

25. The disassembler of Claim 24,

wherein the label address calculating means includes a lower bit calculating unit and an upper bit calculating unit,

the lower bit calculating unit for adding lower bits of the address of the indicated instruction and lower bits of the program counter relative value, setting a result of an addition as lower bits of a label address, and sending any carry generated by the addition to the upper bit calculating unit, and

the upper bit calculating unit adding upper bits of the address of the indicated instruction, upper bits of the program counter relative value, and any carry received from the lower bit calculating unit, and setting a result of the an addition as upper bits of the label address.

26. The disassembler of Claim 24,

wherein the label address calculating means includes a lower bit calculating unit and an upper bit calculating unit,

the lower bit calculating unit adding lower bits of the address of the indicated instruction and lower bits of the program counter relative value without generating a carry, and setting a result of an addition as lower bits of a label address, and

the upper bit calculating unit adding upper bits of the address of the indicated instruction and upper bits of the program counter relative value, and setting a result of an addition as upper bits of the label address.

27. The disassembler of Claim 24, wherein

the label address calculating means adds upper bits of the address of the indicated instruction and upper bits of the program counter relative value, sets a result of an addition as upper bits of the label address, and sets lower bits of the program counter relative value as lower bits of the label address.

28. A debugger that receives an indication of an address of an instruction in object code and replaces the instruction at the indicated address with a replacement instruction, each address of an instruction in the object code having upper bits that indicate a memory address at which a processing packet is stored and lower bits that indicate a position of processing target instruction that is included in the processing packet,

the debugger comprising:

processing packet reading means for reading a processing packet that is indicated by upper bits of the indicated address from the memory and writing the processing packet into an instruction buffer;

instruction writing means for writing the replacement instruction into the processing packet in the instruction buffer over an instruction that is indicated by the lower bits of the indicated address; and

processing packet writing means for writing the processing packet in the instruction buffer back into the memory after the replacement instruction has been written.

29. A compiler that generates an instruction sequence from source code,

the compiler generating a program counter relative value calculating instruction that is executed by a processor, the program counter relative value calculating instruction being an instruction that performs a calculation using a first value and a program counter relative value and uses a result of the calculation to update the first value, the first value being one of

(a) a value of a program counter stored in a register, and

(b) the value stored in a program counter of the processor,

wherein upper bits of the first value indicate a memory address at which a processing packet is stored, and lower bits of the first value of the program counter indicate a processing target instruction that is included in the processing packet.

30. The compiler of Claim 29,

wherein the processor includes a lower bit calculating unit and an upper bit calculating unit,

the program counter relative value calculating instruction having the lower bit calculating unit perform a lower bit calculation and the upper bit calculating unit perform an upper bit calculation,

the lower bit calculation being an addition using lower bits of the first value and lower bits of the value of the program counter relative value, where a result of the lower bit calculation is set as the lower bits of the first value and any generated carry is sent to the upper bit calculating unit, and

the upper bit calculation being an addition using upper bits of the first value, upper bits of the value of the program counter relative value and any carry received from the lower bit calculating unit, where a result of the upper bit calculation is set as the upper bits of the first value.

31. The compiler of Claim 29,

wherein the processor includes a lower bit calculating unit and an upper bit calculating unit,

the program counter relative value calculating instruction having the lower bit calculating unit perform a lower bit calculation and the upper bit calculating unit perform an upper bit calculation, the lower bit calculation being an addition using lower bits of the first value and lower bits of the value of the program counter relative value that does not generate a carry, where a result of the lower bit calculation is set as the lower bits of the first value, and

the upper bit calculation being a calculation using upper bits of the first value and upper bits of the value of the program counter relative value, where a result of the upper bit calculation is set as the upper bits of the first value.

32. The compiler of Claim 29,

wherein the processor includes an upper bit calculating unit,

the program counter relative value calculating instruction having the upper bit calculating unit perform an upper bit calculation and setting lower bits of the program counter relative value as lower bits of the first value, and

the upper bit calculation being an addition using upper bits of the first value and upper bits of the value of the program counter relative value, where a result of the upper bit calculation is set as the upper bits of the first value.

33. A computer-readable recording medium storing an

instruction sequence optimizing program that generates optimized code from an instruction sequence, the instruction sequence optimizing program including:

an address assigning step for estimating a size of each instruction in the instruction sequence and assigning an address to each instruction, upper bits of each address indicating a memory address at which a processing packet is stored and lower bits of each address indicating a processing target instruction in the processing packet;

a label detecting step (1) for detecting a label, which should be resolved by an address of a specified instruction, from the instruction sequence, and obtaining the address of the specified instruction, and

(2) for detecting a label, which should be resolved by a difference in addresses of two specified instructions, from the instruction sequence, and obtaining the addresses of the two specified instructions;

a program counter relative value calculating step for calculating, when a label which should be resolved by a difference in addresses of two specified instructions has been detected, a program counter relative value by subtracting an address of one of the two specified instructions from an address of another of the two specified instructions;

a converting step

(1) for converting an instruction that has a label

that should be resolved by an address of a specified instruction into an instruction with a size that is based on a size of the address of the specified instruction,

(2) for converting an instruction that has a label that should be resolved by a difference in addresses of two specified instructions into an instruction with a size that is based on a size of the program counter relative value calculated from the addresses of the two specified instructions; and

an optimized code generating step for generating optimized code by converting addresses of instructions in accordance with the sizes of instructions after conversion in the converting step.

34. The computer-readable recording medium of Claim 33,

wherein the program counter relative value calculating step includes a lower bit subtracting substep and an upper bit subtracting substep,

the lower bit subtracting substep subtracting lower bits of the address of the one of the two specified instructions from lower bits of the address of the other of the two specified instructions, for setting a result of a subtraction as lower bits of the program counter relative value, and sending any carry generated in the subtraction to the upper bit subtracting substep, and

the upper bit subtracting substep subtracting upper bits of the address of one of the two specified instructions and any carry received from the lower bit subtracting substep from upper bits of the address of the other of the two specified instructions, and for setting a result of a subtraction as upper bits of the program counter relative value.

35. The computer-readable recording medium of Claim 33,

wherein the program counter relative value calculating step includes a lower bit subtracting substep and an upper bit subtracting substep,

the lower bit subtracting substep subtracting lower bits of the address of one of the two specified instructions from lower bits of the address of the other of the two specified instructions without generating a carry and setting a result of a subtraction as lower bits of the program counter relative value, and

the upper bit subtracting substep subtracting upper bits of the address of one of the two specified instructions from upper bits of the address of the other of the two specified instructions, and for setting a result of a subtraction as upper bits of the program counter relative value.

36. The computer-readable recording medium of Claim 33,

wherein the program counter relative value calculating step subtracts upper bits of an address of one of the two specified instructions from upper bits of an address of the other of the two specified instructions, sets a result of a subtraction as upper bits of the program counter relative value, and sets lower bits of the other of the two specified instructions as lower bits of the program counter relative value.

37. A computer-readable recording medium storing an assembler program that generates relocatable code from optimized code that have been generated from an instruction sequence, each address of an instruction in the optimized code having upper bits that indicate a memory address at which a processing packet is stored and lower bits that indicate a position of processing target instruction that is included in the processing packet,

the assembler program comprising:

a label detecting step for detecting a label in the instruction sequence that should be resolved by a difference in addresses between two specified instructions, and obtaining the addresses of the two specified instructions;

a program counter relative value calculating step for calculating a program counter relative value by subtracting an address of one of the two specified instructions from an address of another of the two specified instructions; and

a replacing step for replacing the label with the program counter relative value calculated by the program

~~counter relative value calculating step.~~

~~38. The computer-readable recording medium of Claim 37,~~

~~wherein the program counter relative value calculating step includes a lower bit subtracting substep and an upper bit subtracting substep,~~

~~the lower bit subtracting substep subtracting lower bits of the address of the one of the two specified instructions from lower bits of the address of the other of the two specified instructions, for setting a result of a subtraction as lower bits of the program counter relative value, and sending any carry generated in the subtraction to the upper bit subtracting substep, and~~

~~the upper bit subtracting substep subtracting upper bits of the address of one of the two specified instructions and any carry received from the lower bit subtracting substep from upper bits of the address of the other of the two specified instructions, and for setting a result of a subtraction as upper bits of the program counter relative value.~~

~~39. The computer-readable recording medium of Claim 37,~~

~~wherein the program counter relative value calculating step includes a lower bit subtracting substep and an upper bit subtracting substep,~~

~~the lower bit subtracting substep subtracting lower bits of the address of one of the two specified~~

instructions from lower bits of the address of the other of the two specified instructions without generating a carry and setting a result of a subtraction as lower bits of the program counter relative value, and

the upper bit subtracting substep subtracting upper bits of the address of one of the two specified instructions from upper bits of the address of the other of the two specified instructions, and for setting a result of a subtraction as upper bits of the program counter relative value.

40. The computer-readable recording medium of Claim 37,

wherein the program counter relative value calculating step subtracts upper bits of an address of one of the two specified instructions from upper bits of an address of the other of the two specified instructions, sets a result of a subtraction as upper bits of the program counter relative value, and sets lower bits of the other of the two specified instructions as lower bits of the program counter relative value.

41. A computer-readable recording medium storing a linker program that generates object code from relocatable code that has been generated from an instruction sequence, each address of an instruction in the optimized code having upper bits that indicate a memory address at which a processing packet is stored and lower bits that indicate a

position of processing target instruction that is included in the processing packet,

the linker program comprising:

a relocation information detecting step for detecting a label in the relocatable code that should be resolved by a difference in addresses between two specified instructions, and obtaining the addresses of the two specified instructions;

a program counter relative value calculating step for calculating a program counter relative value by subtracting an address of one of the two specified instructions from an address of another of the two specified instructions; and

a replacing step for replacing the label with the program counter relative value calculated by the program counter relative value calculating step.

42. The computer-readable recording medium of Claim 41,

wherein the program counter relative value calculating step includes a lower bit subtracting substep and an upper bit subtracting substep,

the lower bit subtracting substep subtracting lower bits of the address of the one of the two specified instructions from lower bits of the address of the other of the two specified instructions, for setting a result of a subtraction as lower bits of the program counter relative value, and sending any carry generated in the subtraction to the upper bit subtracting substep, and the upper bit subtracting substep subtracting upper bits of the address of one of the two specified instructions and any carry received from the lower bit subtracting substep from upper bits of the address of the other of the two specified instructions, and for setting a result of a subtraction as upper bits of the program counter relative value.

43. The computer-readable recording medium of Claim 41,

wherein the program counter relative value calculating step includes a lower bit subtracting substep and an upper bit subtracting substep,

the lower bit subtracting substep subtracting lower bits of the address of one of the two specified instructions from lower bits of the address of the other of the two specified instructions without generating a carry and setting a result of a subtraction as lower bits of the program counter relative value, and

the upper bit subtracting substep subtracting upper bits of the address of one of the two specified instructions from upper bits of the address of the other of the two specified instructions, and for setting a result of a subtraction as upper bits of the program counter relative value.

44. The computer-readable recording medium of Claim 41,

wherein the program counter relative value

~~calculating step subtracts upper bits of an address of one of the two specified instructions from upper bits of an address of the other of the two specified instructions, sets a result of a subtraction as upper bits of the program counter relative value, and sets lower bits of the other of the two specified instructions as lower bits of the program counter relative value.~~

~~45. A computer-readable recording medium storing a compiler program that generates an instruction sequence from source code,~~

~~the compiler program generating a program counter relative value calculating instruction that is executed by a processor, the program counter relative value calculating instruction being an instruction that performs a calculation using a first value and a program counter relative value and uses a result of the calculation to update the first value, the first value being one of~~

~~(a) a value of a program counter stored in a register, and~~

~~(b) the value stored in a program counter of the processor,~~

~~wherein upper bits of the first value indicate a memory address at which a processing packet is stored, and lower bits of the first value of the program counter indicate a processing target instruction that is included in the processing packet.~~

46. The computer-readable recording medium of Claim 45,

wherein the processor includes a lower bit calculating unit and an upper bit calculating unit,

the program counter relative value calculating instruction having the lower bit calculating unit perform a lower bit calculation and the upper bit calculating unit perform an upper bit calculation,

the lower bit calculation being an addition using lower bits of the first value and lower bits of the value of the program counter relative value, where a result of the lower bit calculation is set as the lower bits of the first value and any generated carry is sent to the upper bit calculating unit, and

the upper bit calculation being an addition using upper bits of the first value, upper bits of the value of the program counter relative value and any carry received from the lower bit calculating unit, where a result of the upper bit calculation is set as the upper bits of the first value.

47. The computer-readable recording medium of Claim 45,

wherein the processor includes a lower bit calculating unit and an upper bit calculating unit,

the program counter relative value calculating instruction having the lower bit calculating unit perform a lower bit calculation and the upper bit calculating unit perform an upper bit calculation,

the lower bit calculation being an addition using lower bits of the first value and lower bits of the value of the program counter relative value that does not generate a carry, where a result of the lower bit calculation is set as the lower bits of the first value, and

the upper bit calculation being a calculation using upper bits of the first value and upper bits of the value of the program counter relative value, where a result of the upper bit calculation is set as the upper bits of the first value.

48. The computer-readable recording medium of Claim 45,

wherein the processor includes an upper bit calculating unit,

the program counter relative value calculating instruction having the upper bit calculating unit perform an upper bit calculation and setting lower bits of the program counter relative value as lower bits of the first value, and

the upper bit calculation being an addition using upper bits of the first value and upper bits of the value of the program counter relative value, where a result of the upper bit calculation is set as the upper bits of the first value.

ADD B3

Add C2